GP-Based Method for Extracting Exons from DNA Sequence

Takehiro OHTA[1], Ikuo YOSHIHARA[2], Kunihito YAMAMORI[3], Moritoshi YASUNAGA[4]

Abstract

This paper describes GP-based method for extracting exons from DNA sequences. The advantage of GP method is automatically to build a proper model for identification. We made experiments to identify the edges of exon (exon-intron boundary and intron-exon boundary) in DNA sequences and compare the average identification rate by GP method with that by conventional methods (Genscan, HMMgene, Genie, FGENES, Morgan, GeneMark.hmm and MZEF) with data set 'HMR195'. As for exon-intron boundary, GP method is superior to all the conventional methods and as for intron-exon boundary, GP method is superior to almost all the conventional methods except Genscan.

Key Words :
Genetic Programming, Identification, Exon, Splice site

1. Introduction

Human genome consists of 3 billion base pairs. It is divided into several kinds of regions, for example exon, intron, promoter and so on. Exons are the regions translated into protein and occupy about 5% in all the base sequence of human genome. Introns and others not translated into protein[1-2].

Large amount of research have been performed for extracting exons. Hidden Markov Model (HMM), Neural Network (NN) and so on are used widely for identifying exon regions, and they can identify with high accuracy. However, in case of HMM, it is difficult to design the number of state and state transitions properly. In the case of NN, it is difficult to design proper structure of the neural network. To avoid the matter, we develop a Genetic Programming (GP)-based method for extracting exon. The advantage of using GP is automatically to build proper model.

We make experiments to identify exon-intron boundaries and intron-exon boundaries in DNA sequences and compare the average identification rate by GP method with that by conventional methods with data set 'HMR195'.

2. Boundaries Between Exons and Introns

2.1 Exon-intron boundary and intron-exon boundary

DNA consists of four kinds of bases; A (Adenine), C (Cytosine), G (Guanine) and T (Thymine). The first two letters of introns are mostly 'GT' and the last two letters are 'AG' (Fig.1). An exon is sandwiched by 'AG' and 'GT' of introns. In this paper, we take up only boundaries with this feature. Identifying the edges of exon ('GT' and 'AG') is necessary to extract exon. 'GT' and 'AG' form the edges of an exon, but they exist not only on the boundary but also inside the exons and introns. The number of 'GT's and 'AG's on the boundary is far smaller than that on non-boundary. We make experiments to identify whether 'GT's and 'AG's is boundary or not by our method.

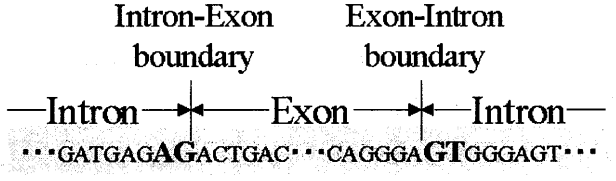

Fig.1 The boundary of exon and intron

1) Graduate Student, Dept. of Computer Science and Systems Engineering, Miyazaki University

2) Professor, Dept. of Computer Science and Systems Engineering, Miyazaki University

3) Associate Professor, Dept. of Computer Science and Systems Engineering, Miyazaki University

4) Associate Professor, Institute of Information Sciences and Electronics, University of Tsukuba

2.2 Genome data

A large amount of human genome data are released on the web site of NCBI (National Center for Biotechnology Information), from where we can obtain experimental data[3]. This site shows if 'GT's and 'AG's is boundary or not.

After searching 'GT' or 'AG' from base sequences, we take out 10 bases from upstream and 10 bases from downstream of 'GT', and take out 30 bases from upstream and 7 bases from downstream of 'AG' (Fig.2). The length of sequence is decided based on the analysis of appearance frequency of bases near 'GT's and 'AG's[4-5].

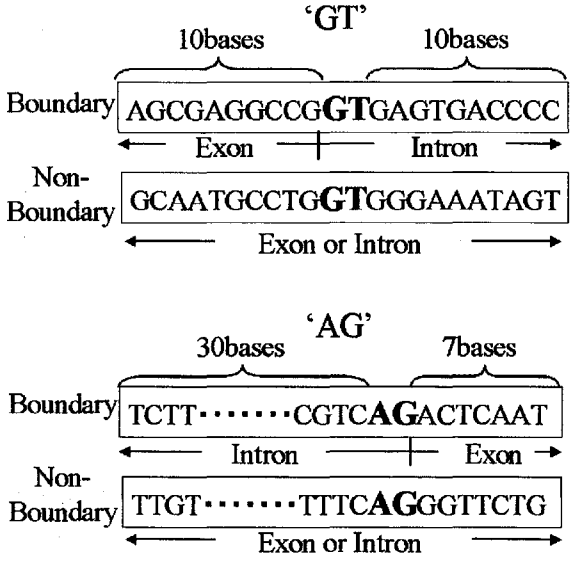

Fig.2 Sub-sequences near 'GT' and 'AG'

3. An Automatic Model Building Method Using Genetic Programming

3.1. Binary coding of base sequences data

Four kinds of bases are represented by binary strings (1: True, 0: False) as follows:

$$A = (0\,0\,0\,1)_2, \quad C = (0\,1\,0\,0)_2,$$
$$G = (0\,0\,1\,0)_2, \quad T = (1\,0\,0\,0)_2.$$

Learning signals are given 1 (True) for boundary and 0 (False) for non-boundary.

Fig.3 shows an example of binary coding of exon-intron boundary. We regard 22-letter base sequences as 88-bit string, and use the bit string as the input to the identification model. (Fig.2).

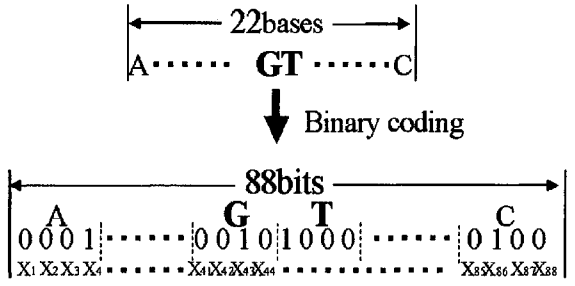

Fig.3 Binary coding of exon-intron boundary

3.2 Genetic coding

The model for identifying boundaries is expressed by tree structure (Fig.4). Internal nodes of the tree are logical operators (Table.1) and leaves are input data. Root of the tree outputs 1 (True) for boundary or 0 (False) for non-boundary[6].

'M(2,3)' and 'NM(2,3)' are the operators that have 3 inputs and 1 output. 'M(2,3)' decides output by a majority of inputs.

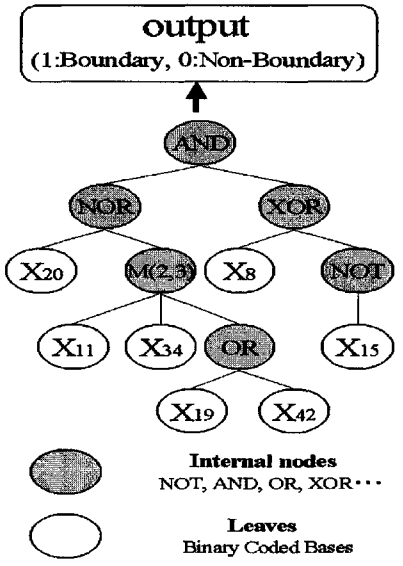

Fig.4 Tree structure expression of identification model

Table.1 Logical operators

The number of arguments	Logical operator
1	NOT NOP (No-operation)
2	AND OR XOR NAND NOR XNOR
3	M(2,3) (Majority decision) NM(2,3) (NOT M(2,3))

3.3 Genetic Operations

The procedure of automatic model building by GP is shown in Fig.5[7-10].

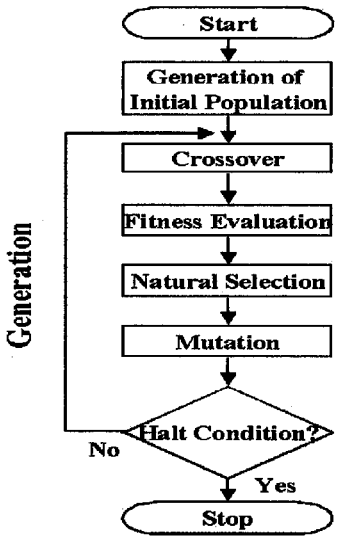

Fig.5 Flow chart of model building by GP

(1) Generation of Initial Population

Random-structured trees are generated as initial population.

(2) Crossover

Two individuals are selected at random from the population as a pair of parents. They exchange their sub-trees each other and breed children. Fig.6 shows an example of crossover to produce two children. After crossover both parents are still alive and population size temporarily expands.

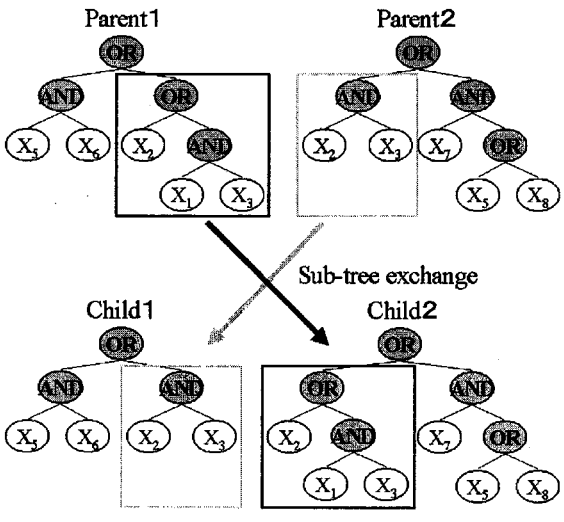

Fig.6 Example of crossover

(3) Fitness Evaluation

The fitness is defined as follows:

$$fitness = \frac{1}{E}, \tag{1}$$

$$E = \sum_{i=1}^{t} (\tilde{Y}_i - Y_i)^2 + h(m), \tag{2}$$

$$h(m) = Cm, \tag{3}$$

here, t: the number of data for model building,

C: positive number, m: the number of nodes.

The first term of Eq.(2) is the number of errors, and 2nd is penalty $h(m)$. Penalty is introduced for limiting the complexity of the tree structure.

(4) Natural Selection

Ranking selection is employed for natural selection. Individuals are ranked according to fitness, and chosen from the highest fitness up to the population size.

(5) Mutation

Individuals ranked under top two are mutated with a predefined probability. To keep top two individuals aims at averting to destroy good characteristics of the population. Mutation is change of logical operator with the same number of arguments (Fig.7).

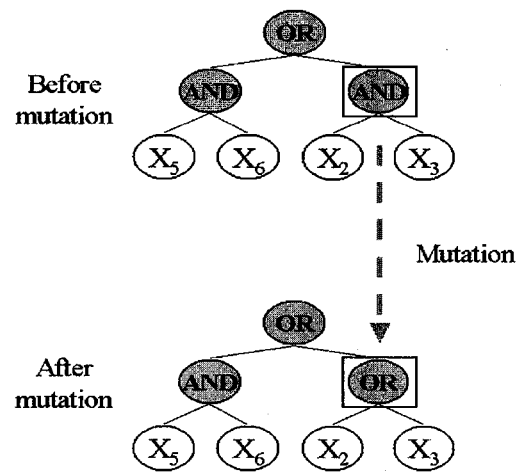

Fig.7 Example of mutation

4. Identification of Exon-Intron Boundary and Intron-Exon Boundary

4.1 Experimental conditions

The amount of human genome sub-strings we collected from NCBI is as follows.

Exon-intron boundary ('GT')	
- Boundary	2100
- Non-boundary	20000
Intron-exon boundary ('AG')	
- Boundary	2100
- Non-boundary	20000
Total number of data	44200

For model building, we randomly choose 1000 boundary data and 1000 non-boundary data. Other 20100 data are used for validation. We make 10 data sets based on above-rules, and experiment 10 times per each data set, so the total number of experiments is 100 times.

The parameters of GP are as follows.

Maximum number of nodes	100
Population size	100
Mutation rates	25%
Crossover rates	100%
Maximum generation	4000

4.2 The measure of accuracy of identification rate

Sensitivity(Sn) and Specificity(Sp) are widely used to measure the accuracy of identification of exon-intron boundary and intron-exon boundary. Sn is the percentage of truly identified boundaries among the true one. Sp is the percentage of truly identified boundaries among all those identified boundaries.

$$\text{Sensitivity (Sn)} = \frac{b}{B} \times 100 \quad (\%) \tag{4}$$

$$\text{Specificity (Sp)} = \frac{b}{b + n'} \times 100 \quad (\%) \tag{5}$$

In our research, we define Sensitivity'(Sn') to evaluate the identification rate of non-boundary.

$$\text{Sensitivity' (Sn')} = \frac{n}{N} \times 100 \quad (\%) \tag{6}$$

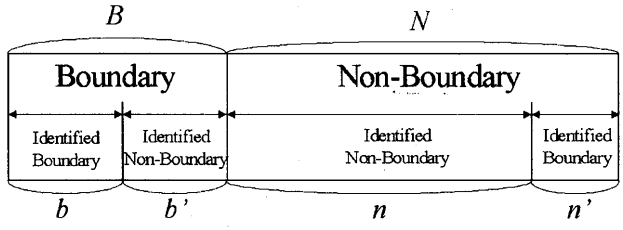

Fig.8 Result of identification

4.3 Identification rate by GP

Table.2 shows identification rate by GP with the human genome benchmark data. As for exon-intron boundary, the average identification rate by GP is 91.8% for boundary and 90.0% for non-boundary. As for intron-exon boundary, the average identification rate by GP is 84.7% for boundary and 78.7% for non-boundary.

Table.2 Identification rate by GP

	Identification Rate (%)	
	GT	AG
Boundary(Sn)	91.8	84.7
Non-Boundary(Sn')	90.0	78.7

4.4 Comparing GP with conventional methods

We compare GP with conventional methods in regard to Sn of exon-intron boundary and intron-exon boundary. The conventional methods for comparison are famous tools for genome analysis: Genscan, HMMgene, Genie, FGENES, Morgan, GeneMark.hmm and MZEF[11]. Methods used in conventional tools are shown in Table.3.

Table.3 Conventional methods and their analysis method

Conventional methods	Analysis method
Genscan	Generalized hidden markov model
HMMgene	Hidden markov model
GeneMark.hmm	Hidden markov model
Genie	Markov model
Morgan	Decision tree, Dynamic programming
FGENES	Linear discriminant analysis
MZEF	Quadratic discriminant analysis

Benchmark data is HMR195 that consists of DNA base sequence of human, mouse and rat. The amount of data in HMR195 is as follows.

Exon-intron boundary ('GT')
- Boundary 740
- Non-Boundary 20000

Intron-exon boundary ('AG')
- Boundary 740
- Non-Boundary 20000

Total number of data 41480

For model building, we randomly choose 370 boundary data and 370 non-boundary data. Other 20000 data are used for validation.

Fig.9 and Fig.10 show comparison of GP and conventional methods in regard to exon-intron boundary and intron-exon boundary respectively. As for exon-intron boundary, GP method is superior to all the conventional methods and as for intron-exon boundary, GP method is superior to almost all the conventional methods except Genscan.

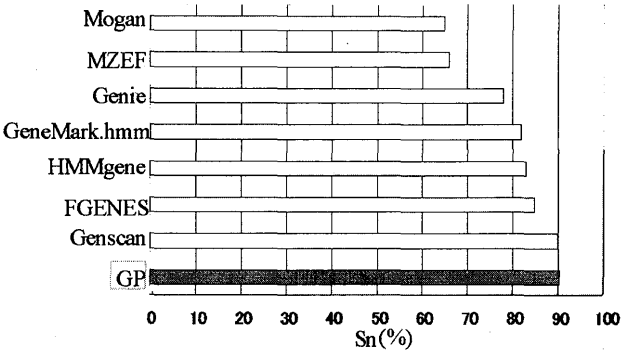

Fig.9 Comparison of identification rate ('GT')

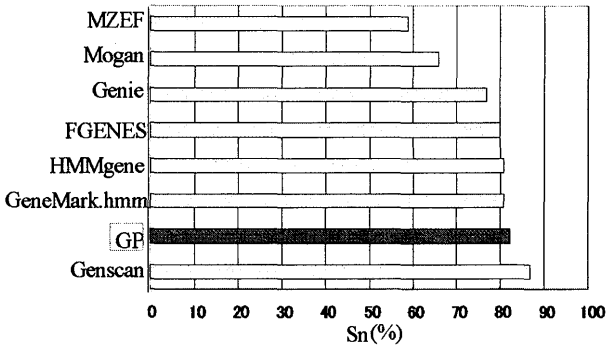

Fig.10 Comparison of identification rate ('AG')

5. Conclusion

We developed GP-based method for extracting exons from DNA sequence. The advantage of our method is automatically to build proper model for identification.

We made experiments to identify exon-intron boundaries and intron-exon boundaries in DNA sequences and compared the average identification rate by GP method with that by conventional methods with data set 'HMR195'.

As for exon-intron boundary, average identification rate of GP is 91.8% for boundary and 90.0% for non-boundary. As for intron-exon boundary, average identification rate of GP is 84.7% for boundary and 78.7% for non-boundary.

As for exon-intron boundary, GP method is superior to all the conventional methods and as for intron-exon boundary, GP method is superior to almost all the conventional methods except Genscan.

6. Acknowledgement

This research is supported in part by the 2002 grant of the Ministry of Education, Culture, Sports, Science and Technology in Japan (Grant No. 14015206).

References

1) Kanehisa, M., "Invitation to Genome Information", KYORITSU SHUPPAN CO., LTD., 1996, (in Japanese)

2) Kanehisa, M., *et al.*, "Human Genome Projects", KYORITSU SHUPPAN CO., LTD., 1997, (in Japanese)

3) National Center for Biotechnology Information <URL>http://www.ncbi.nlm.nih.gov/

4) Yoshihara, I., Kamimai, Y. and Yasunaga, M., "Multi-Modal neural network for Identifying Exon-Intron Boundaries", KES '01, IOS Press, 2001, pp.998-1002

5) Yoshihara, I., Kamimai, Y. and Yasunaga, M., "Feature Extraction from Genome Sequences Using Multi-Modal Neural Network", Genome Informatics 2001, Universal Academy Press, Inc., 2001, pp.420-422

6) Ohta, T., Yoshihara, I., Yamamori, K. and Yasunaga, M., "GP-based Method for Identifying Exon Region in DNA sequences", Proc. of the 4th Asia-Pacific Conf. on Simulated Evolution and Learning (SEAL'02), cr1333:CD-ROM, 2002

7) Yoshihara, I., Numata, M., Aoyama, T., Yasunaga, M. and Abe, K.: "Extending Prediction Term of GP-based Time-series Model", Proc. of the 5th Int. Symp. on Artificial Life and Robotics (AROB-V), 2000, pp. 268-271

8) Numata, M., Sugawara, K., Yoshihara, I. and Abe, K.: "Time Series Prediction by Genetic Programming", 3rd Annual Genetic Programming Conference (GP-98), 1998, pp.176-179

9) Ohta, T., Yoshihara, I., Yamamori, K. and Yasunaga, M.,: "Development of a Time Series Prediction method by Genetic Programming", SICE2001, pp.469-474

10) Banzhaf, W., Nordin, P., Keller, R., and Francone, F.: "Genetic Programming — An Introduction", Morgan Kaufmann Publishers, Inc., 1998

11) Rogic, S., Mackworth, A. and Ouellette, F.: "Evaluation of Gene-Finding Programs on Mammalian Sequences", Genome Research vol11, 2001, pp.817-832